

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
30 August 2001 (30.08.2001)

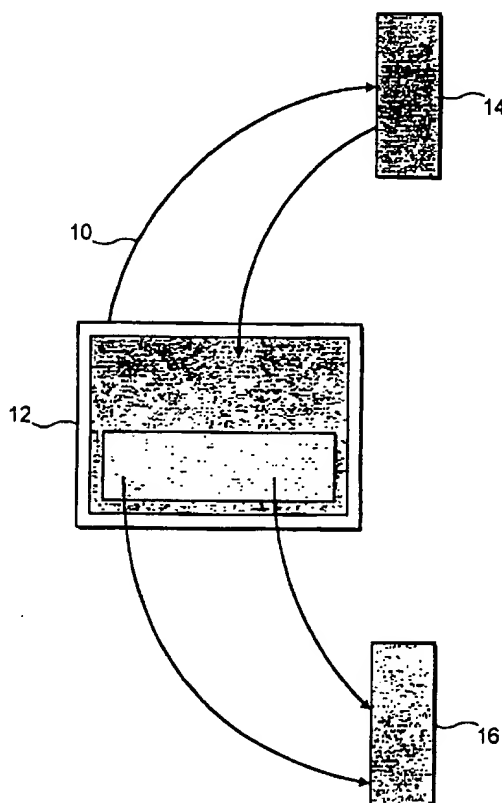
PCT

(10) International Publication Number
WO 01/63878 A1

- (51) International Patent Classification⁷: **H04L 29/06** [GB/GB]; 268 Leek Road, Endon, Staffordshire ST9 9BQ (GB).
- (21) International Application Number: PCT/GB01/00754
- (22) International Filing Date: 22 February 2001 (22.02.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
00301409.9 23 February 2000 (23.02.2000) EP
0004304.2 23 February 2000 (23.02.2000) GB
- (71) Applicant (for all designated States except US): **TRADE-SAFELY.COM LIMITED** [GB/GB]; Manchester House, 86 Princess Street, Manchester M1 6MG (GB).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): **HAWKES, Michael**
- (74) Agent: **LLOYD, Patrick, Alexander, Desmond**; Reddie & Grose, 16 Theobalds Road, London WC1X 8PL (GB).
- (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR INTERNET WEB SITE ACCREDITATION



(57) Abstract: To authenticate an Internet web site, the web site is verified each time it is accessed by a user browser (12). On receipt of a page request, the web site generates a web page into which is embedded a URL site address or other unique identifier. A cookie, link to an image or java applet or similar, for execution of the verification process may also be embedded or may be resident on the user browser. At the user, the browser instantiates the cookie, link to image or applet and extracts the URL site address or unique identifier which is sent, together with user ID to a verification server (16). The server performs comparison, comparing the URL address or unique identifier with a record of the correct identifier and/or URL address. Based on the comparison, an approval status is generated and sent back to the browser. This is displayed as a graphic in the browser or in any other way. The graphic includes an animated portion to increase security. The graphic also includes a user generated code or challenge which is stored on the verification server and which is sent back to the user upon successful authentication.

WO 01/63878 A1

WO 01/63878 A1



Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD AND APPARATUS FOR INTERNET
WEB SITE ACCREDITATION

FIELD OF THE INVENTION

This invention relates to Internet web site
5 accreditation; that is the verification that a site on the
World Wide Web (www) is bona fide.

BACKGROUND OF THE INVENTION

In recent years, e-commerce, the business of trading
across the Internet has grown very rapidly and growth is
10 forecast to continue growing rapidly. This growth has
brought with it problems in security and authenticity. Many
web sites exist which purport to be the genuine source for
goods but which are actually bogus, either selling
counterfeit goods of being an unauthorised source for
15 genuine goods.

It is presently very easy for a bogus third party to
register an Internet domain name which is identical or very
similar to a well known brand. Internet search engines will
find these similar IP addresses and take browsers to any
20 site bearing a close resemblance in name or page content.
Thus is the correct, legitimate address is
bloggsandbloke.com it is very common to find unrelated
addresses such as bloggs+bloke.com, bloggs-bloke.com,
bloggsandbloke.co.uk etc.

25 It is equally easy to lift the front page of the
genuine web site of bloggsandbloke.com and import it onto
the rogue site. As result, browsers, and their users, think
that they are in the genuine site of Bloggs and Bloke PLC,
when, in reality, they are visiting a rogue site.

30 In the past attempts have been made to authenticate web
sites to give customers a degree of security. However, all
authentication systems have been static; a given web site is
authenticated in advance and can display an authentication

- 2 -

symbol, for example, an authentication logo, typically on its home page. The authentication symbol is displayed automatically every time that site is accessed. Whilst such systems offer a degree of consumer protection, there are far from infallible. Not only can authentication symbols be copied, the accreditation cannot take into account changes made to a web site's content or the site provider. If the front page has been copied the unlawful copies will also include the authentication.

Security mechanisms already exist to allow safe (encrypted) transfer of information such as credit card numbers etc. However, these systems merely provide an encryption mechanism and offer no guarantee as to the authenticity of the site.

There is, therefore, a need for an improved system for accreditation and authentication of Internet web sites. The invention aims to meet this need and, in its broadest form, provides a method of dynamic authentication in which authentication takes place each time the site is accessed by a remote user.

In its broadest form, the invention provides a method, apparatus and programme which ensures that the web site is verified by a verification server each time it is accessed. More specifically there is provided a method of verifying the authenticity of an Internet web site comprising the steps of: sending a page request from a user to the site to be authenticated; at the site to be verified, generating a web page containing a site identification and sending the generated page to the user; at the user, extracting the site identification and sending it to a verification server; at the verification server, comparing the site identification with a pre-stored identification; and based on the comparison, indicating to the user whether or not the site is authentic, the indication including sending a predetermined user originated identifier back to the user.

- 3 -

The invention further provides a system for verifying the authenticity of an Internet web site comprising: at the site to be authenticated: means for generating a web page on receipt of a page request from a user browser, the
5 generating means including means for embedding in the page a site identification; and means for sending the generated page to the user browser; at the user: means for extracting the site identification; and means for sending it to a verification server; at the verification server: means for
10 comparing the site identification with a pre-stored identification; and means for, indicating to the user whether or not the site is authentic, based on the comparison, and, if authentic, returning a predetermined user originated identifier to the user.

15 The invention further provides a computer program comprising program code means for performing, when the program is run on a computer or a computer network, the steps of: on receipt of a page request from a remoter location, generating a web page containing a site
20 identification and sending the generated page to the remote location; at the remote location, extracting the site identification and sending it to a verification server; receiving from the verification server at the remote location an indication of whether or not the site is
25 authentic based on a comparison at the verification server of the site identification and a pre-stored identification; displaying at the remote location an identifier originating from the remote location if the site is authentic; and displaying at the remote location the indication received
30 from the verification server.

Embodiments of the invention have the advantage that the front page of a web site cannot be lifted in such a manner that it carries the verification as the verification is generated every single time the web site is accessed and
35 the user receives a user originated identifier which ensures that the verification signal originates from the authenticating server.

- 4 -

Preferably, the web page generated by the site to be verified has embedded within in it a unique identification number whereby the web page is unique. This has the advantage that the page will not be cached by an Internet Service Provider. Internet Service Providers generally only
5 cache the most frequently visited sites as a means of cutting down response times. As the site is then not cached the authentication process cannot be bypassed by obtaining pages from the cache.

10 Preferably, the indication to the user of whether or not the site is authentic comprises a graphic display within the browser, or at least a portion of the browser. Preferably at least a part of this graphic is animated. Animation of a portion of the graphic makes it harder for
15 the graphic to be copied.

The web page generated by the site to be authenticated, may contain an applet or a cookie or a link to the verification server, which may be resident on that user's browser. The applet or cookie or link mechanism performs the
20 functions of extracting site identification from the generated web page once it is received at the user's browser, communicating with the verification server, sending the verifying server a code or a challenge phrase, created by the user and generating the graphic display with the
25 challenge phrase which is sent back to the user.

Preferably the applet handles each of the three operations mentioned above as separate concurrent threads.

BRIEF DESCRIPTION OF THE DRAWINGS

An embodiment of the invention will now be described,
30 by way of example, and with reference to the accompanying drawings in which:

Figure 1 is a schematic view of a system embodying the invention;

35 Figures 2a and 2b are a flow chart illustrating the process embodying the invention;

- 5 -

Figure 3 is a block diagram illustrating the functionality of the client applet or cookie; and

Figure 4 is a flow chart illustrating the procedure executed by the applet or cookie.

5 The system (10), illustrated in figure 1 comprises a user browser (12), which may be a conventional Internet browser such as Microsoft Internet Explorer or Netscape Navigator which may be modified to include a custom Java applet or cookie which will be described. The browser may
10 run on any convenient platform such as a PC or Macintosh. The browser communicates with a corporate server 14 over a conventional Internet link via an Internet service provider (ISP) (not shown). In order to access the corporate server, which hosts the corporation's web site, the browser sends
15 the conventional hypertext transfer Protocol (HTTP) address for the server and sends a page request. The authentication system to be described is based on the concept of the corporate server generating a fresh web page with a unique serial number and a java applet or cookie embedded in it.
20 This web page is received at the user browser and the applet or cookie causes a verification request to be sent to authentication server 16 which is remote from the corporate server and which verifies the URL address and signals back to the user browser that the site has either been verified
25 or that the verification has failed. The applet or cookie may be resident on the browser or may be downloaded each time from the corporate server embedded in the uniquely generated web page.

 The system described enables verification that data
30 received from a web site has come from an accredited source. Thus, the data delivered is provided by a certified data provider and the accreditation status is verified each time the web site is accessed. Once the site has been verified, the results of the verification process are made known to
35 the Internet browser in a dynamic format, preferably a multi-layer dynamic format to make logo theft difficult.

- 6 -

Furthermore, when the user communicates with verifying server, he sends a code or challenge phrase which then accompanies the verification logo, signs or signal when it is sent back to the user.

5 At each stage of the process, measures are included to prevent ISP caching mechanisms. This ensures that authentication is performed each time the site is accessed. For this reason, and as an additional safety measure, a unique serial number is provided by the corporate server a
10 value which is unique and generated each time a page request is received.

 The system will now be described in greater detail. The Internet browser (12) preferably uses one of two common protocols for information interchange: HTTP and CGI. Other
15 protocols may be used. HTTP (Hypertext Transfer Protocol) is a simple to use protocol in which requests to a server are made in standard text and the initial responses are text only. The protocol is simple to implement and debug.

 Common Gateway Interface (CGI) is a well known protocol
20 used to generate composite documents. HTTP is a protocol for delivering of documents which have no real processing capabilities. To increase the usability of the World Wide Web, server-side programs are run that generate page content when required. These are often written in simple scripting
25 languages. Data is passed to these server scripts via HTTP through the CGI. The server scripts convert incoming information from CGI standard text format into a more usable form using well known procedures.

 It is common practice for ISP's to prove a local copy
30 of popular web sites. The ISP then responds to page requests as if they had originated from the original provider. This caching process speeds up access to these sites' information. It will be understood that in the preferred embodiment of the invention, caching of authenticated sites
35 is not possible as the method requires the actual corporate server to generate a unique code each time the site is accessed. Thus, a dynamic page is generated containing the

- 7 -

authentication applet with each page being unique. The ISP is unable to cache the HTTP page request and the applet is therefore delivered from the publisher, the corporate server, rather than the ISP. Caching does not take place as
5 the web page is unique and therefore not recognised by the ISP as one which is frequently accessed. This is also the case if the applet is resident on the browser.

To increase security still further, metatags may be provided in the corporate page to prevent attempts at
10 caching by the ISP.

The system outlined above may also incorporate a system for the re-authorisation of pre-authorised web sites each time the site is accessed. The pre-authorisation step is an off-line process performed by the verification authority.
15 When the owner of the corporate server, the publisher, has satisfied the verifying authority that they are a genuine goods or service provider within the criteria to be set by the verifying authority, the publisher is registered and issued with a unique serial number and a script to generate
20 the remaining data which, together with the serial number will be embedded in the web page generated when the corporate server is accessed. These two elements of the page serial number for a unique request number which will be delivered to the applet as a routine parameter.

25 Once established, the unique serial number is embedded in the page sent to the user browser as a parameter to the applet. The unique serial number may take the form:

<supplied serial number> - <UTC based integer> - [<CGI process ID or iteration count>].

30 An example of a unique identifier code is as follows:
Supplied Serial number (registration no): PDQ4567X
Current UTC time value when page requestion: 949424629
Current CGI process ID (or iteration count): 6541
The unique number generated is: PDQ4567X-949424629-
35 6541.

This number provides the verifying authority with two pieces of verification data. The first is the registered

- 8 -

publisher serial number which had been supplied by the verifying authority to the publisher, which is used as the primary key when performing site verification. If verification fails, the verification server can identify the source of the errant applet.

The optional second verification data is the time, according to the publisher, that the applet was delivered. This is stored at the verification server cache as the last time the applet was used. If a request is made with a substantially older time stamp, a security exception is raised and the request considered stale. Where a time stamp is only marginally older it will still be accepted as the routing path could have been more complex and a delayed legitimate request. The time is expressed as a Universal Time Code.

The iteration count is a number which automatically increments each time a page request is received and can wrap around at the maximum number of concurrent processes supported by the publisher's system architecture. Some systems cannot support an iteration count and a CGI process ID may be used instead each process ID number is unique. This may be used, for example, in UNIX based systems.

Where a request is stale, the applet forces a page reload from the original publisher to overcome local browser cache or user-connection problems.

The process will now be described in more detail with reference to figure 2. In the following description, it is assumed that the site to be accessed has already been assigned a serial number by the verifying authority.

At step 100, a page delivery script in CGI Script is sent from the Internet browser (12), to the corporate web-server (14).

The web-server receives the page delivery request and performs the steps necessary to generate and send a unique web page back to the browser. At step 102 the web server receives the page request. At step 104 the unique serial number is generated comprising the provided serial number,

- 9 -

the UTC date and time and the iteration count or process id as described earlier.

At step 106 the corporate web server sets a codebase for Java applet or cookie delivery. Applets or cookies may be used interchangeably in this embodiment of the invention. As the applet is the more complex, it will be described in detail. The manner in which the invention is implemented using cookies will be described later. The codebase is the absolute network location from which the applet or cookie is delivered and, in the case of the applet, forms part of the applet program. It ensures that secure connection may be made to the verification server.

At step 108 the corporate web server creates a web page in which is embedded the applet and the unique serial number. The applet will be described in greater detail later on.

At step 110 the newly created web page is sent to the user's Internet browser.

Thus at this stage, the user has sent a page request to the corporate web server which has replied with a web page which is unique to that request.

In practice, the user has merely clicked on a hypertext link to the corporation owning the corporate web server or entered the site's URL in the address field of the browser. The user's browser will not yet display anything from the corporate web server.

At step 112, the browser receives the web page from the corporate web server and at step 114 instantiates the applet embedded in the page. The applet then executes within the Java (or similar) environment.

At step 116 the Java applet extracts the document base for the current page. This is the absolute location, that is the network address and the page location of the document delivered over the World Wide Web. The (optional) embedded code base is then retrieved from the delivered page and the current user environment is noted. This latter process records the user network address for reporting.

- 10 -

Java (or similar mechanisms) support concurrent operation with a number of tasks running at the same time within its own environment. Each of these tasks is an execution thread. At step 118, the applet creates two new
5 concurrent execution threads. New thread 1 handles graphics operations and new thread 2 handles communications with the verification server.

At step 120 thread 1 of the applet draws a background graphic frame. At step 122 thread 2 of the applet opens a
10 secure connection to the verification server 16. At step 124 thread 2 sends the page serial number, network delivery, page addresses and current environment data to the verification server. This data may be sent in encrypted form. At step 126 thread 1 of the applet creates pending and
15 clipping frames. The latter is a frame which defines an area of an image which requires updating. Any drawing requests falling outside that defined area will be ignored.

At step 128 thread 1 of the applet overlays the background, pending and clipping frames and at step 130
20 displays the composite frame and starts animation. The user will now see the composite frame displayed in their browser. While thread 1 of the applet is executing, the verification server, at 132 receives and accepts the secure connection request. At step 134 the request is analysed and a serverlet
25 instantiated to enable multiple concurrent verification transactions. It will be appreciated that the system described considers only one user and one corporate server. In practice, the verification server will handle requests from many users relating to many corporate servers. The
30 serverlet is a small Java based program running within the verification net server.

At step 136 the serverlet decodes the received data and stores it for later analysis and security exception reports. At step 138 the serverlet performs a Java Data Base
35 Connectivity (JDBC) lookup to validate the data. Depending on the results of the lookup, the serverlet either generates an approved status with an embedded serial number at step

- 11 -

140 or a not approved status. In the latter case a non approved status is generated if the corporate server has an unknown address or the (optional) serial number is not available, or where the serial number is stale. Where the
5 system is not approved a security exception is generated. At step 142 the serverlet returns the approval status to the user's Internet browser. At step 140 the serverlet closes and at step 142 the connection to the verification web server closes.

10 At step 144 the user's Internet browser receives the approval status and at step 146 thread 2 of the applet generates a 'pass' or 'fail' graphic pane. If a 'pass' is displayed, the (optional) serial number is also displayed. This pane overrides the pending frame becoming the middle
15 layer of the animated logo. At step 148 thread 2 closes any active terminals, releases resources and terminates. At step 146 thread 1 enters an animation loop until the browser moves to another page or closes. The animation is displayed in the area defined by the clipping frame and may be a logo
20 indicating that the corporate site is verified as authentic. At step 152 the applet releases all resources when the browser moves to another page or closes. The threads are stopped and any memory used is cleared. The applet then terminates.

25 The Java applet is illustrated in more detail in figure 3. As discussed in the foregoing, the applet comprises three threads: thread 0, thread 1 and thread 2. When the applet is first instantiated thread 0 is created automatically on the Java Virtual Machine which is resident on all web browsers
30 within the browser environment. It instantiates the other two execution threads and creates the necessary components for the other threads to manipulate. The thread eventually loops through a graphic refresh request and a small 100ms pause becoming the animation controller.

35 When the applet receives a termination request, thread 0 stops all other threads and the releases all resources before terminating itself (step 152, figure 2).

- 12 -

At step 202 thread 0 requests a logo component from the verification server and at step 204 the logo is drawn onto the clipping frame. Of course, the logo is only drawn if the site has been verified.

5 Thread 1, at step 300 creates three graphics frames. The thread is responsible for creating or updating the system graphics double buffer such that the applet is self animating. The graphics sub-system has three layers: a base frame which contains a shape such as a background rectangle,
10 background and static imagery; a middle frame containing any moving imagery; and a top, clipping frame which limits the drawing area to a defined area as discussed previously. The three layers are re-assembled and repainted as one graphic image which becomes the image source for thread 0. Thus, in
15 figure 3, the three layers are drawn at step 302 and overlayed onto the base graphic at step 304. The middle layer is updated at step 304, for example to show a rotating graphic and the current verification status. The final two steps are looped until the browser is closed or moves to a
20 new page.

 The third thread, thread 2, handles client-server communications and acquires the information required to validate a page. The thread replaces the middle frame used by thread 1 with an updated graphic once the main
25 verification server has sent its response back to the applet. Thus, in figure 3 at step 400 thread 2 opens a connection to the verification server. At step 402 the unique page number, the delivery address and browser data are sent to the verification server and at step 404 the
30 response is decoded and, according to the response, a new middle frame is generated. At step 406 the middle frame generated by thread 1 is replaced with the new graphic.

 The verification server performs a database look up using JDBC to ensure that the details provided by the
35 corporate server match. As the UTC and iteration count/process id are unique it is only the supplied serial number which is compared along with the codebase. If any

- 13 -

inconsistencies are found the server sends a fail status back to the applet. If none are found a success status is returned.

Communications with the server are over a secure
5 communication service which is well-known. As a new connection is established the server spawns a new task to deal with the incoming request, ensuring that each task cannot interfere with any other task and providing a mechanism to balance server loading over multiple back end
10 machines.

Data inconsistencies or processing difficulties automatically generate a security exception. These are analysed at the verification server. Pattern matching techniques may be used to identify hacking activity. As each
15 applet has an individual signature, misuse of an applet can be traced and an offending web or hacking site can be tracked.

In order to understand the applet process further, the pseudocode for the process running on the browser is as
20 follows, where the reference numerals refer to figure 4.

START - 500

CHECK FOR PREVIOUS DETAILS - 502

IF PREVIOUS DETAILS NOT FOUND THEN

INPUT CHALLENGE PHRASE - 504

25 REGISTER CHALLENGE PHRASE WITH SERVER - 506

ENDIF

START ANIMATION THREAD

START COMMUNICATION THREAD

LOOP UNTIL STOP SIGNAL RECEIVED

30 REDISPLAY ANIMATION THREAD FRAME

IF POINTER WITHIN APPLET BOUNDS THEN

- 14 -

```

                                DISPLAY FLOATING CHALLENGE PHRASE
                                ENDIF
                                ENDLOOP

                                END

5      COMMUNICATION THREAD:

                                START
                                EXTRACT PAGE DETAILS AND BROWSER ENVIRONMENT - 508
                                SEND PAGE DETAILS AND BROWSER ENVIRONMENT TO SERVER

                                AWAIT SERVER RESPONSE - 510

10     IF TIMEOUT OR SERVER VERIFICATION FAILURE THEN
                                CALL VERIFICATION FAILURE IN ANIMATION THREAD
                                ELSE
                                GET SERIAL NUMBER AND CHALLENGE PHRASE FROM RESPONSE
                                CALL VERIFICATION SUCCESS IN ANIMATION THREAD
15     ENDIF

                                STOP - 518

                                ANIMATION THREAD

                                START
                                CREATE ANIMATION GRAPHIC FRAMES (3) - 512
20     CREATE BACKGROUND IN FRAME (0)
                                CREATE PENDING LOGO IN FRAME (1)
                                CREATE CLIPPING WINDOW IN FRAME (2)

                                LOOP:      - 516
                                DRAW FRAME (0)
25     OVERLAY FRAME (1)
                                OVERLAY FRAME (2)

                                UNTIL STOP SIGNAL RECEIVED

                                ANIMATION THREAD:VERIFICATION FAILURE: 514
                                START
30     CREATE FAILURE IMAGE IN FRAME (1)
                                END

                                ANIMATION THREAD:VERIFICATION SUCCESS:
                                START
                                CREATE VERIFICATION IMAGE IN FRAME (1)
35     END
```

- 15 -

As mentioned above, cookies and link mechanisms may be used in place of applets. Cookies and link mechanisms perform the process described in a similar manner except that most of the processing is carried out on the verification server. Thus, in figure 2, at step 108, link mechanism is embedded in the web page sent to the User's browser and at steps 114, 116 and 118 the web browser executes, extracts the document base and creates the concurrent execution threads respectively. The generation of the verification sign or logo and the accompanying challenge phrase are handled differently. Both are received from the verification server, as an image without having to be generated by an applet.

The manner in which challenge phrases are used in the embodiments described will now be described in greater detail. It will be appreciated from the discussion of the browser pseudo-code and figure 4 that a challenge phrase is registered with the verification server by the user when they first use the system. This is performed when the user signs up or subscribes to the verification system. The user is required to provide a challenge phrase; a code which may be viewed as a user originated identifier chosen by him. This is sent to the verification server and stored and the user is registered.

When an authentication is performed, the unique web page downloaded from the merchant includes an image tag, for example in HTML form. This may be in the form:

```

```

The user's browser downloads the image tags from the merchant. The user browser then generates a link to the verification server, the verification server asks a first time user to register giving his name and user defined

- 16 -

challenge phrase or identifier. The verification server stores the user data and sends a cookie containing a unique ID to the user browser.

In cases of users already registered with the verification
5 server, The user's browser downloads the image tags from the merchant. The user browser then generates a link to the verification server, the verification server looks at the merchants address (URL) or site identification and compares this with pre stored data. If the verification server
10 recognises the address (URL) and / or site identification, it will generate an authentication signal to the user. The verification server also checks the user ID which is extracted from the cookie stored on the user's browser, and attaches the user defined identifier or challenge phrase to
15 the authentication signal. This enables the user to be confident that verification has been properly performed by the correct verification server.

At the verification server, the referring page's URL is extracted, then the cookie having the user's unique ID is
20 extracted. The verification server compares the referring URL address with a database of approved URL addresses and searches a database of registered users for the extracted user ID number and extracts the challenge phrase associated with it. The verification server then sends, preferably as
25 an image, a response to the user which includes the challenge phrase.

- 17 -

Claims

1. A method of verifying the authenticity of an Internet web site comprising the steps of:
 - 5 sending a page request from a user to the site to be authenticated;
 - at the site to be verified, generating a web page containing a site identification and sending the generated page to the user;
 - at the user, extracting the site identification and
 - 10 sending it to a verification server;
 - at the verification server, comparing the site identification with a pre-stored identification; and
 - based on the comparison, indicating to the user whether, or not the site is authentic, the indication including
 - 15 sending a predetermined user originated identifier back to the user.
2. A method according to claim 1, wherein the site identification includes a serial number and the address from which the generated web page was sent.
- 20 3. A method according to claim 1 or 2, wherein the site identification includes a unique identifier which is embedded in the generated web page to form a unique web page.
4. A method according to claim 3, wherein the unique
- 25 identifier includes an iteration count.
5. A method according to claim 3, wherein the unique identifier includes a process identification.
6. A method according to claim 4, 5 or 6, wherein the unique identifier includes a universal date code.

- 18 -

7. A method according to any preceding claim, wherein the step of sending the site identification to the verification server comprises sending an encrypted communication over a secure connection.
- 5 8. A method according to any preceding claim, comprising embedding an applet in the generated web page.
9. A method according to any of claims 1 to 7, comprising embedding a cookie in the generated web page.
- 10 10. A method according to claim 8, comprising including links to an image.
11. A method according to any of claims 1 to 10, wherein the unique user ID is transmitted to the verification server.
- 15 12. A method according to any preceding claim, wherein the user pre-registers with the verification site and submits at pre-registration the user originated identifier.
- 20 13. A method according to any preceding claim, wherein the steps of indicating to the user whether or not the site is authentic comprises generating an approved or not approved status and sending the generated status to the user browser.
14. A method according to any preceding claim, wherein the indication from the verification server is accompanied by the user originated identifier.
- 25 15. A system for verifying the authenticity of an Internet web site comprising:
at the site to be authenticated:
means for generating a web page on receipt of a page request from a user browser, the generating means including means for embedding in the page a site identification; and

- 19 -

means for sending the generated page to the user browser;

at the user:

means for extracting the site identification; and

5 means for sending it to a verification server;

at the verification server:

means for comparing the site identification with a pre-stored identification; and

10 means for, indicating to the user whether or not the site is authentic, based on the comparison, and, if authentic, returning a predetermined user originated identifier to the user.

16. A computer program comprising program code means for performing, when the program is run on a computer or a computer network, the steps of:

15 on receipt of a page request from a remoter location, generating a web page containing a site identification and sending the generated page to the remote location;

20 at the remote location, extracting the site identification and sending it to a verification server;

receiving from the verification server at the remote location an indication of whether or not the site is authentic based on a comparison at the verification server of the site identification and a pre-stored identification;

25 displaying at the remote location an identifier originating from the remote location if the site is authentic; and

displaying at the remote location the indication received from the verification server.

30 17. A computer program product comprising program code means stored on a computer readable medium for performing, when run on a computer or a computer network, the steps of:

35 on receipt of a page request from a remote location, generating a web page containing a site identification and sending the generated page to the remote location;

- 20 -

at the remote location, extracting the site identification and sending it to a verification server;

receiving from the verification server at the remote location an indication of whether or not the site is authentic based on a comparison at the verification server
5 of the site identification and a pre-stored identification;

displaying at the remote location an identifier originating from the remote location if the site is authentic; and

10 displaying at the remote location the indication received from the verification server.

18. A computer program or program product according to claim 16 or 17, wherein the site identification includes a serial number and the address from which the generated web
15 page was sent.

19. A computer program or program product according to claim 16, 17 or 18, wherein the site identification includes a unique identifier which is embedded in the generated web page to form a unique web page.

20 20. A computer program or program product according to claim 19, wherein the unique identifier includes a universal date code, and either an iteration count or a process iteration count.

21. A computer program or program product according to any
25 of claims 16 to 20, wherein the step of generating a web page includes embedding an applet in the web page.

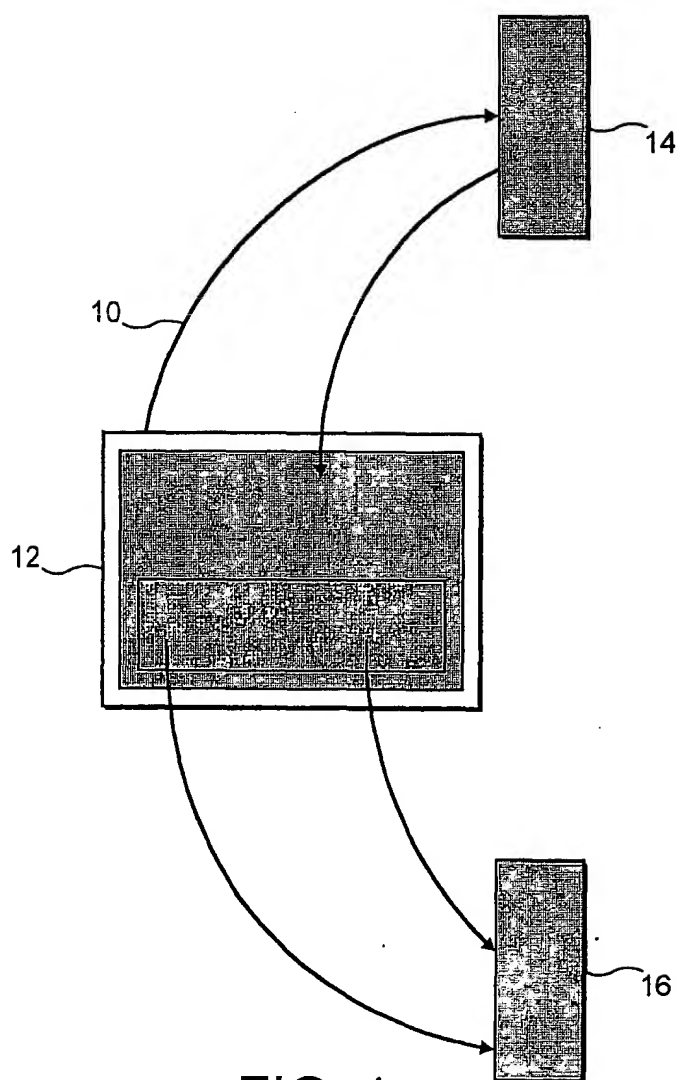


FIG. 1

2 / 5

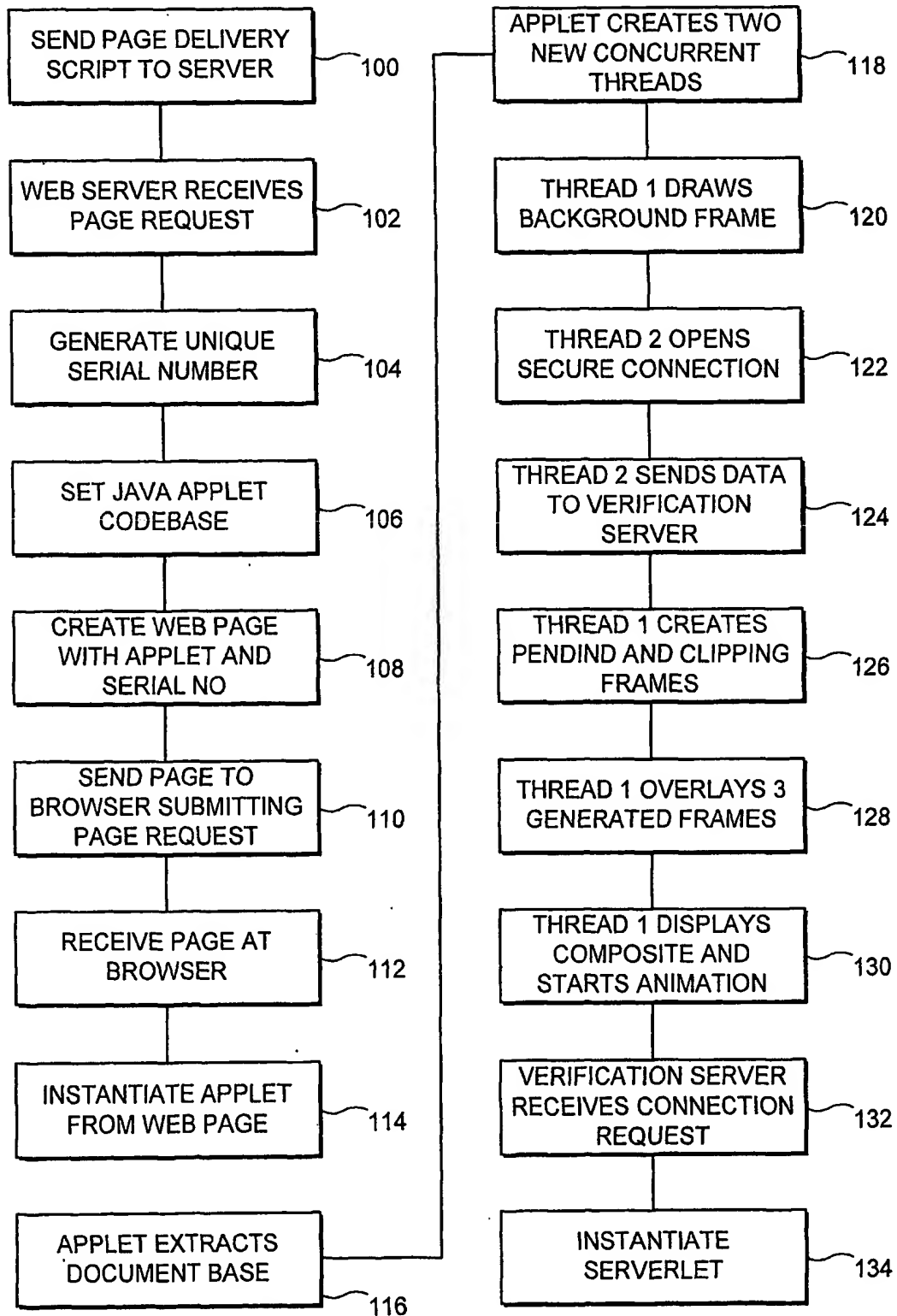


FIG. 2a

SUBSTITUTE SHEET (RULE 26)

3 / 5

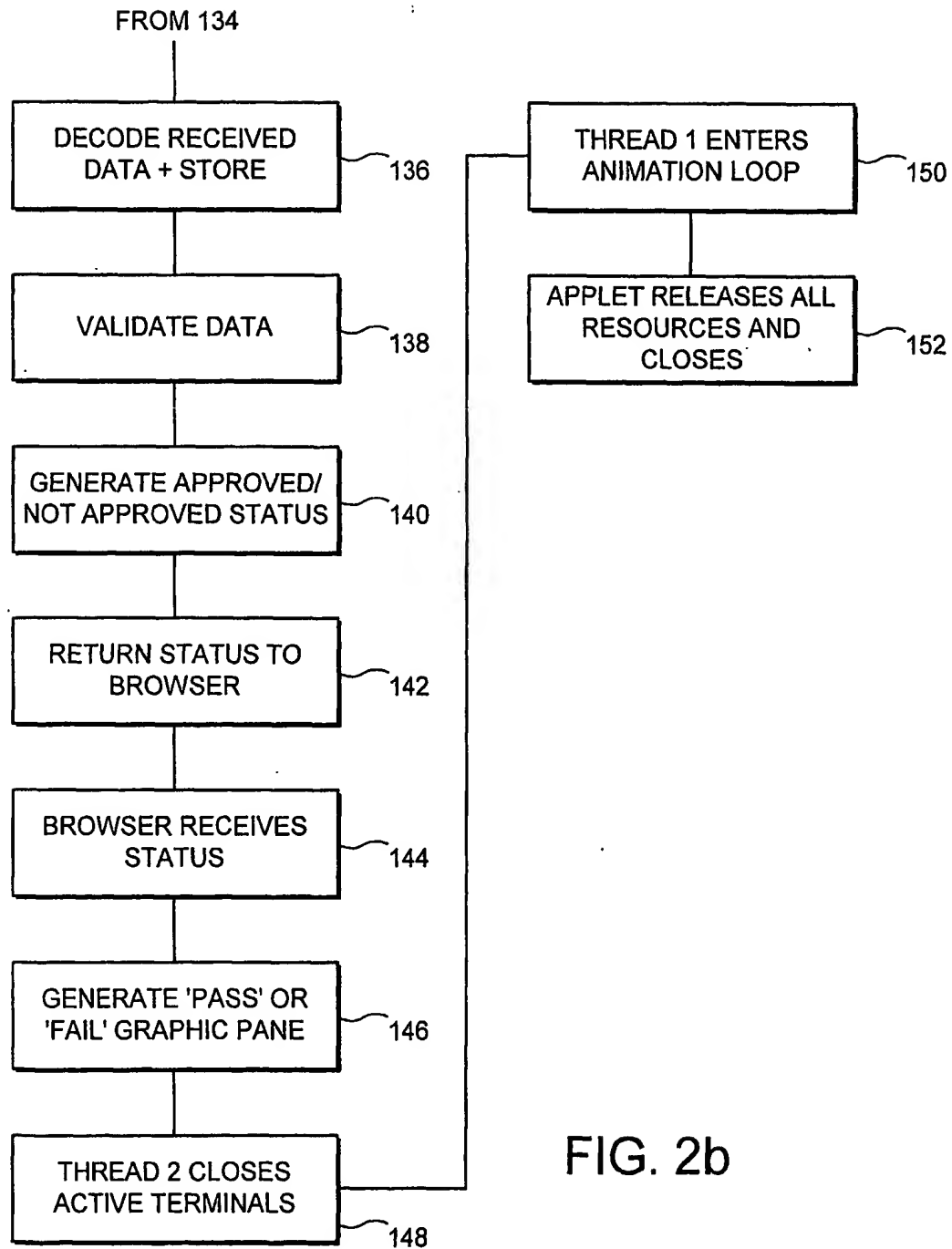


FIG. 2b

4 / 5

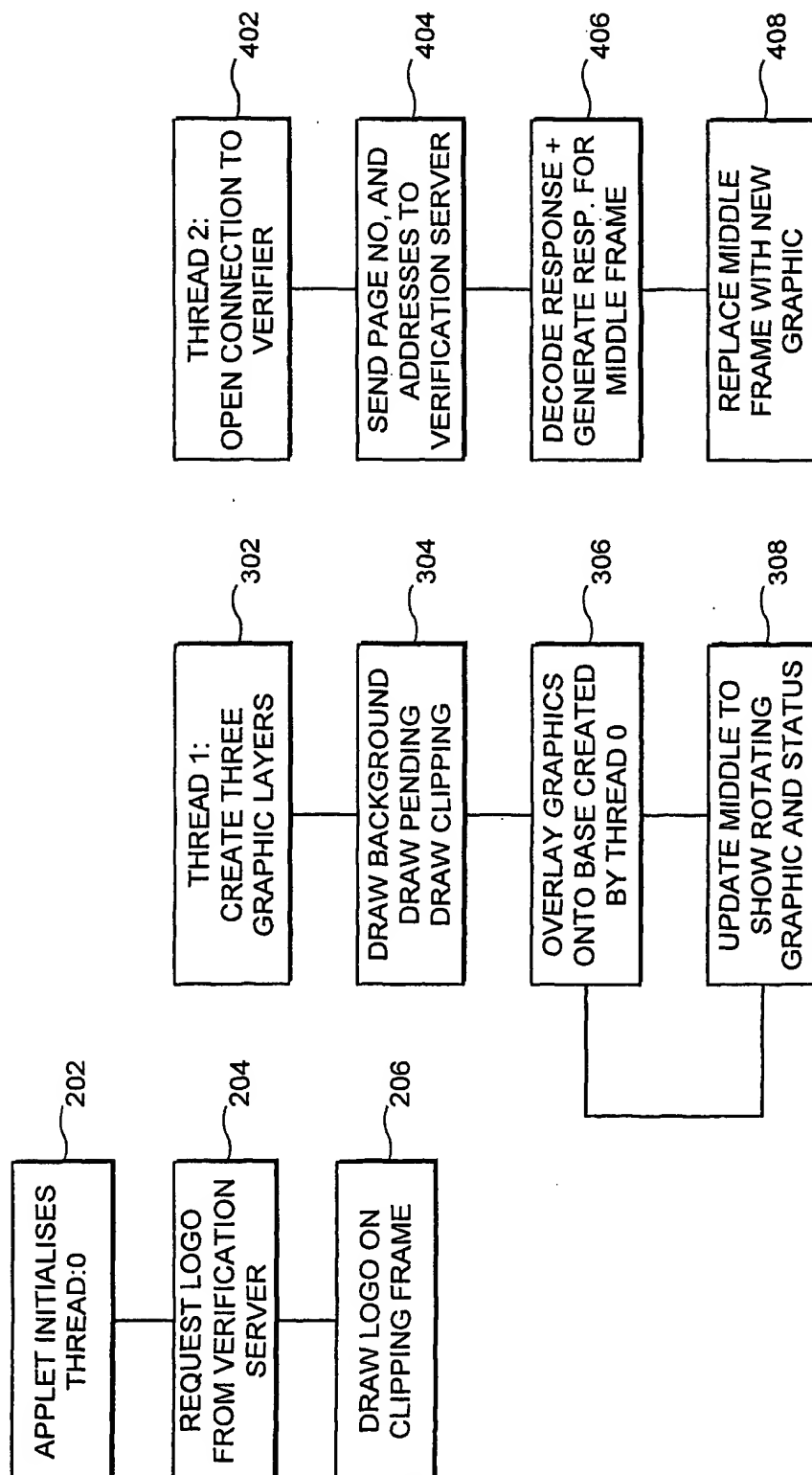


FIG. 3

5 / 5

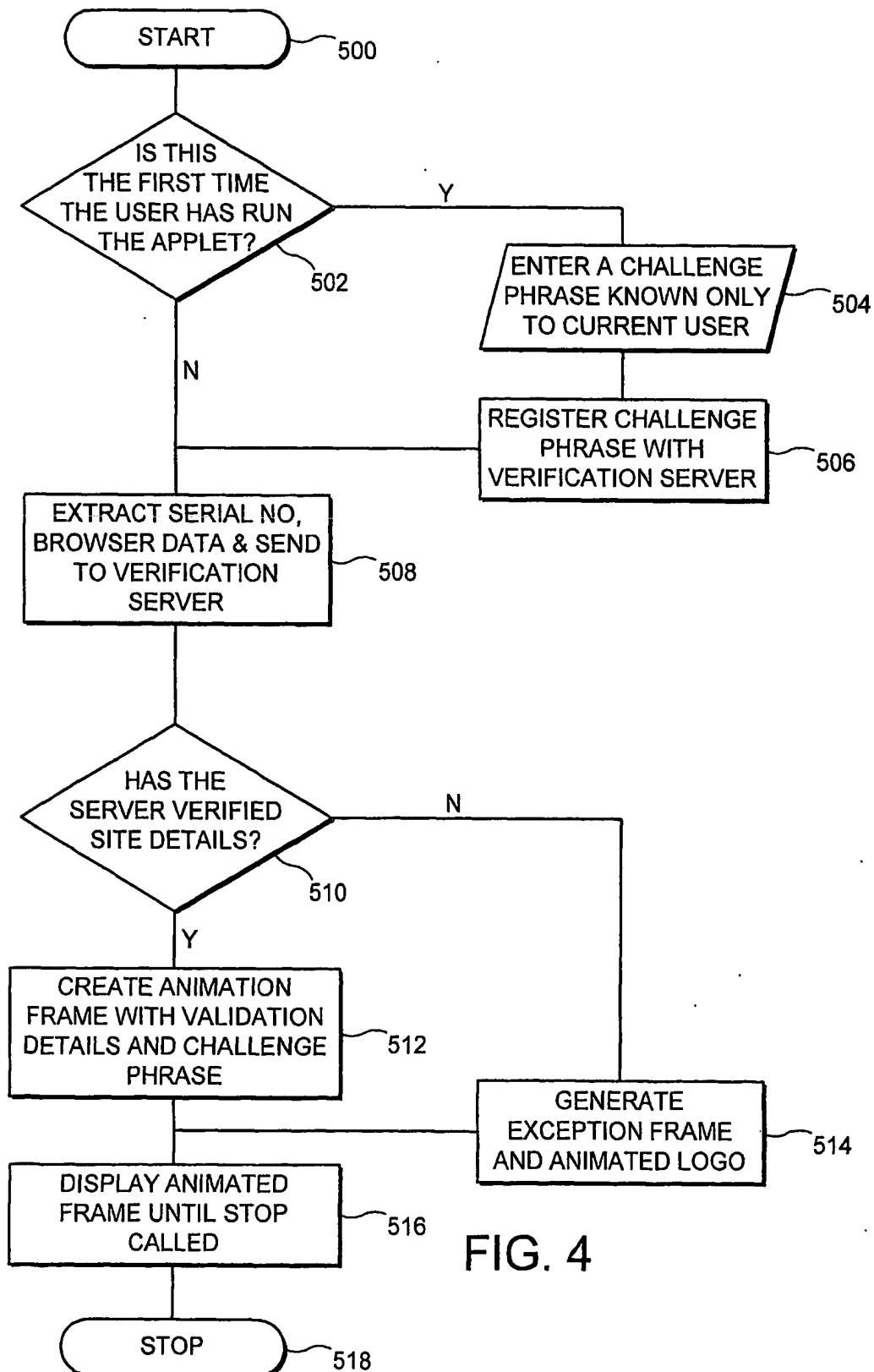


FIG. 4

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 01/00754

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>US 6 018 801 A (PALAGE M D ET AL) 25 January 2000 (2000-01-25) column 1, line 6 - line 13 column 1, line 66 - column 2, line 21 column 2, line 48 - column 3, line 9 column 4, line 22 - line 33 column 5, line 25 - line 35 column 6, line 14 - line 65 column 7, line 30 - line 42 column 8, line 30 - column 9, line 10</p> <p style="text-align: center;">-/-</p>	1-21

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- *A* document member of the same patent family

Date of the actual completion of the international search

1 June 2001

Date of mailing of the international search report

11/06/2001

Name and mailing address of the ISA
European Patent Office, P.B. 5816 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax (+31-70) 340-3016

Authorized officer

Vercauteren, S

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 01/00754

C(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 936 531 A (HITACHI LTD) 18 August 1999 (1999-08-18) paragraph '0010! paragraph '0012! - paragraph '0013! paragraph '0015! paragraph '0020! paragraph '0024! paragraph '0032! -----	1-21

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int'l. Application No

PCT/GB 01/00754

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
US 6018801	A	25-01-2000	EP	0990350 A	05-04-2000
			WO	9943167 A	26-08-1999
EP 0936531	A	18-08-1999	JP	11232286 A	27-08-1999